//à terminer

Titre : Gérer les retards

Acteurs principaux : Employé

Acteur secondaire :

Préconditions :

Post-conditions : Le registre des articles en retard a été mise-à-jour avec le montant de l’amande dû.

Scénario Principal

|  |  |
| --- | --- |
| Acteurs | Système |
| 1. //Initialisation du scénario à déterminer  2. L’employé lance la gestion des retards  et choisi les paramètres de l’étendu du rapport | 3. Le système affiche la liste des articles en retards, la durée, l’adhérent et l’amende.  4. Le système imprime le résultat de la requête |

Scénario alternatif

|  |  |
| --- | --- |
| Acteurs | Système |
|  | 3a. Il n’y a aucun article en retard  3.1 Le système informe l’employé qu’il n’y a aucun article en retard. Le scénario prend fin |

Contrats :

|  |  |
| --- | --- |
| **Opération** | chercherRetards(numéroAdhérent) : ResultSet |
| **Préconditions** | Un objet de type RapportRetards a été créé. |
| **Postconditions** | Un objet res de type ResultSet a été créé et est associé à une requête comprenant les articles dont la date de retour précède la date courante.  Un objet de type LigneRetard a été créé pour chaque ligne de l’objet res dans RapportRetards.  Si numéroAdhérent n’est pas nul, le ResultSet comprend les lignes de la table ArticlesALouer où ArticlesALouer.Adherent = numéroAdhérent.  L’attribut LigneRetard.Adherent = db.ArticlesALouer.Adherent.  L’attribut LigneRetard.codeArticle = db.ArticlesALouer.ID.  L’attribut LigneRetard.JoursEnRetard = Days.daysBetween(db.ArticlesALouer.DateRetour, now()).getDays();  L’attribut int amende correspond à calculerAmende(int JoursEnRetard, String codeArticle) |
| **Opération** | Imprimer() |
| **Préconditions** | RapportRetard a été initialisé avec des objets ligneRetard. |
| **Postconditions** | Les attibuts String codeArticle, titreFilm, adherent, joursEnRetard et amende de chaque objet ligneRetard est transmise au output en format compréhensible par le client. |